IN THE CLAIMS:

1. (original) A system comprising:

a server coupled to a network;

a network adapter to couple said server to said network, said network adapter having a plurality of adapter ports; and

an adapter driver executing on said server, said adapter driver having a plurality of instances corresponding to said plurality of adapter ports, wherein said adapter driver, in response to a request to change a configuration of a selected instance of said plurality of instances, is to,

determine if there is data flow through the selected instance of said adapter driver, and if not, to

block all subsequent data flow through the selected instance of said adapter driver,

block all subsequent information requests to said adapter driver relating to the selected instance, and

reinitialize said selected instance of said plurality of instances without rebooting said server.

2. (original) The system of claim 1, further comprising a storage coupled to said server via said network adapter, and wherein said network is a storage area network and said network adapter is a Fibre Channel adapter with PCI-X connectivity.

3. (original) The system of claim 2, wherein said server is one of a Solaris™ server and a Windows™ server.

4. (original) The system of claim 1, wherein said adapter driver manages at least said plurality of adapter ports of said network adapter, and wherein each of said plurality of adapter ports are referenced individually on a per-instance basis by said adapter driver.

5. (original) The system of claim 4, wherein said adapter driver further manages a second network adapter having a second plurality of adapter ports, and wherein each of said second plurality of adapter ports are also referenced individually on a per-instance basis by said adapter driver.

6. (original) The system of claim 1, wherein, using a graphical user interface of a driver management application, a user may issue said request to change the configuration of said selected instance of said plurality of instances.

7. (original) The system of claim 6, wherein said request to change the configuration of said selected instance is one of a driver parameter update request, a driver unload/load request, and a request to recognize a new device on said network.

8. (original) The system of claim 7, wherein said driver management application is further to parse a configuration file of said selected instance into a name/value parameter list.

9. (original) The system of claim 8, wherein said driver management application retrieves a previous initialization time stamp for the selected instance, said previous initialization time stamp to indicate the last time said selected instance was initialized.

10. (original) The system of claim 9, wherein said driver management application is further to,

clear a plurality of old driver parameters for said selected instance,

define a plurality of new driver parameters based on said request for the configuration change, and

request that said adapter driver activate said plurality of new driver parameters.

11. (original) The system of claim 10, wherein said driver management application, following said reinitialize of the selected instance of said plurality of instances is to,

request a new initialization time stamp for the selected instance,

compare said new initialization time stamp to said previous initialization time stamp, and

if said new initialization time stamp is later than said previous initialization time stamp, indicate that said reinitialization of said selected instance is successful.

12. (original) A method comprising:

executing an adapter driver on a server that is coupled to a network, said server being coupled to the network using a network adapter, and said adapter driver having a plurality of instances corresponding to a plurality of adapter ports of said network adapter;

receiving a request to change a configuration of a selected instance of said plurality of instances;

determining if there is data flow through the selected instance of said adapter driver;

blocking, if there is no data flow through the selected instance, all subsequent data flow through the selected instance of said adapter driver;

blocking, if there is no data flow through the selected instance, all subsequent information requests to said adapter driver relating to the selected instance; and

reinitializing said selected instance of said plurality of instances without rebooting said server.

13. (original) The method of claim 12, wherein said executing the adapter driver comprises executing the adapter driver on said server that is coupled to said storage area network, said server being coupled to the storage area network using a Fibre Channel network adapter with PCI-X connectivity, and said adapter driver having said plurality of instances corresponding to said plurality of adapter ports of the network adapter.

14. (original) The method of claim 13, wherein said executing the adapter driver comprises executing the adapter driver on said server that is coupled to said storage area network, said server being one of a Solaris™ server and a Windows™ server and being coupled to the storage area network using the Fibre Channel network adapter with PCI-X connectivity, and said adapter driver having said plurality of instances corresponding to said plurality of adapter ports of the network adapter.

15. (original) The method of claim 12, further comprising:

managing, by said adapter driver, at least said plurality of adapter ports of said network adapter; and

referencing said plurality of adapter ports individually on a per-instance basis by said adapter driver.

16. (original) The method of claim 15, further comprising:

managing, by said adapter driver, a second network adapter having a second plurality of adapter ports; and

referencing said second plurality of adapter ports individually on a per-instance basis by said adapter driver.

17. (original) The method of claim 12, further comprising issuing, by a user using a graphical user interface of a driver management application, said request to change the configuration of said selected instance.

18. (original) The method of claim 17, wherein said issuing comprises issuing, by the user using the graphical user interface of said driver management application, said request to change the configuration of said selected instance, wherein said request is one of a driver parameter update request, a driver unload/load request, and a request to recognize a new device on said network.

19. (original) The method of claim 18, further comprising parsing, by said driver management application, a configuration file of said selected instance into a name/value parameter list.

20. (original) The method of claim 19, further comprising retrieving, by said driver management application, a previous initialization time stamp for the selected instance, said previous initialization time stamp to indicate the last time said selected instance was initialized.

21. (original) The method of claim 20, further comprising:

clearing, by said driver management application, a plurality of old driver parameters for said selected instance,

defining, by said driver management application, a plurality of new driver parameters based on said request for the configuration change; and

requesting, by said driver management application, that said adapter driver activate said plurality of new driver parameters.

22. (currently amended) The method of claim 21, further comprising, following said reinitialization of the selected instance of said plurality of instances[[,]]:

requesting, by said driver management application, a new initialization time stamp for the selected instance;

comparing said new initialization time stamp to said previous initialization time stamp; and

indicating that said reinitialization of said selected instance is successful if said new initialization time stamp is later than said previous initialization time stamp.

23. (currently amended) A computer program product comprising:

a computer readable tangible ~~usable~~ medium having ~~computer program code~~ stored ~~embodied~~ therein ~~to display information, the computer program product having~~:

computer readable program code to execute an adapter driver on a server that is coupled to a network, said server being coupled to the network using a network adapter, and said adapter driver having a plurality of instances corresponding to a plurality of adapter ports of said network adapter;

computer readable program code to receive a request to change a configuration of a selected instance of said plurality of instances;

computer readable program code to determine if there is data flow through the selected instance of said adapter driver,

computer readable program code to block, if there is no data flow through the selected instance, all subsequent data flow through the selected instance of said adapter driver;

computer readable program code to block, if there *is* no data flow through the selected instance, all subsequent information requests to said adapter driver relating to the selected instance; and

computer readable program code to reinitialize said selected instance of said plurality of instances without rebooting said server.

24. (original) The computer program product of claim 23, wherein said computer readable program code to execute the adapter driver comprises computer readable program code to execute the adapter driver on said server that is coupled to said storage area network, said server being coupled to the storage area network using a Fibre Channel network adapter with PCI-X connectivity, and said adapter driver having said plurality of instances corresponding to said plurality of adapter ports of the network adapter.

25. (original) The computer program product of claim 24, wherein said computer readable program code to execute the adapter driver comprises computer readable program code to execute the adapter driver on said server that is coupled to said storage area network, said server being one of a Solaris™ server and a Windows™ server and being coupled to the storage area network using the Fibre Channel network adapter with PCI-X connectivity, and said adapter driver having said plurality of instances corresponding to said plurality of adapter ports of the network adapter.

26. (original) The computer program product of claim 23, further having:

computer readable program code to manage, by said adapter driver, at least said plurality of adapter ports of said network adapter; and

computer readable program code to reference said plurality of adapter ports individually on a per-instance basis by said adapter driver.

27. (original) The computer program product of claim 26, further having:

computer readable program code to manage, by said adapter driver, a second network adapter having a second plurality of adapter ports; and

computer readable program code to reference said second plurality of adapter ports individually on a per-instance basis by said adapter driver.

28. (original) The computer program product of claim 23, further having computer readable program code to issue, by a user using a graphical user interface of a driver management application, a request to change a configuration of said selected instance.

29. (original) The computer program product of claim 28, wherein said computer readable program code to issue comprises computer readable program code to issue, by the user using the graphical user interface of said driver management application, said request to change the configuration of said selected instance, wherein said request for the configuration change is one of a driver parameter update request, a driver unload/load request, and a request to recognize a new device on said network.

30. (original) The method of claim 29, further having computer readable program code to parse, by said driver management application, a configuration file of said selected instance into a name/value parameter list.

31. (original) The method of claim 30, further having computer readable program code to retrieve, by said driver management application, a previous initialization time stamp for the selected instance, said previous initialization time stamp to indicate the last time said selected instance was initialized.

32. (original) The method of claim 31, further having:

computer readable program code to clear, by said driver management application, a plurality of old driver parameters for said selected instance,

computer readable program code to define, by said driver management application, a plurality of new driver parameters based on said request for the configuration change; and

computer readable program code to request, by said driver management application, that said adapter driver activate said plurality of new driver parameters.

33. (original) The method of claim 32, further having, following said computer readable program code to reinitialize said selected instance,:

computer readable program code to request, by said driver management application, a new initialization time stamp for the selected instance;

computer readable program code to compare said new initialization time stamp to said previous initialization time stamp; and

computer readable program code to indicate that said reinitialization of said selected instance is successful if said new initialization time stamp is later than said previous initialization time stamp.